IDA

INSTITUTE FOR DEFENSE ANALYSES

Optimum ID Sensor Fusion for Multiple Target Types

J. Kent Haspert

March 2000

Approved for public release;
distribution unlimited.

IDA Document D-2451

Log: H 00-000745

This work was conducted under contract DASW01 98 C 0067, Task CA-2-1617, for the Joint Theater Air and Missile Defense Organization. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

INSTITUTE FOR DEFENSE ANALYSES

IDA Document D-2451

Optimum ID Sensor Fusion for Multiple Target Types

J. Kent Haspert

PREFACE

This document was prepared as part of the work performed under a task entitled "Analyses of CID Technical Capabilities, Procedures, and Data in Support of Joint Theater Air and Missile Defense Joint Mission Area Assessment."

CONTENTS

TABLES

FIGURES

I. INTRODUCTION

Combat identification (CID) is often viewed as the weakest link in the military's kill chain.[1] The CID problem requires good ID sensors, but unfortunately no ID sensor works perfectly for all targets under all conditions. This leads to the employment of multiple types of ID sensors—which in turn introduces the problem of deciding how to combine (or fuse) the information from multiple sensors. Traditional sensor fusion uses fixed rules (i.e., declare a target hostile if at least one ID sensor indicates it is hostile, so long as none says it is a friend). Although these fixed rules are easy for operators to implement, they typically do not lead to the optimum solution. Bayesian techniques do enable optimum ID sensor fusion. However, Bayesian techniques have heretofore treated the ID problem as one involving only two types of targets: friends and hostiles. Today's combat environment often involves additional types of targets (e.g., coalition forces, neutrals). This paper addresses this more complex situation.

The paper begins with a review of basic identification (ID) sensor fusion concepts and the ideas presented in Ralston's paper[2] for the two types of true targets he considered. It then describes the alternative formulation for the two-dimensional (2–D) (i.e., two-target-type) problem. The paper then extends the alternative formulation to three types of targets and then to any number of types of targets. The paper also shows that multiple-target-type ID fusion is a mathematically trivial process to implement, so long as the requisite sensor ID probability values and ID declaration criteria (e.g., the relative costs of correct and incorrect ID decisions) are available.

[1] The kill chain can include search, detect, track, classify, identify, assign, solution of fire control calculations, weapon launch, mid-course guidance, weapon acquisition of the target, terminal homing, fusing, target damage, and kill assessment.

[2] James M. Ralston, "Bayesian Sensor Fusion for Minimum-Cost I.D. Declarations," 1998 Joint Service Combat Identification Systems Conference on Requirements, Technologies, and Developments, (CISC–98), Volume 1—Technical Proceedings.

II. REVIEW OF BAYESIAN ID SENSOR FUSION CONCEPTS

An ID sensor acts on a target and provides two or more outputs. The likelihood that various sensor outputs will occur is a function of the type of target truly being seen. Table 1 illustrates an ID probability matrix for a hypothetical sensor. This hypothetical sensor is assumed to provide only three types of outputs: a friend indication (F_i), a hostile indication (H_i), and an unknown indication (U_i). A good ID sensor will tend to have large probability values in cells A and E and small values in the other cells.

Table 1. Hypothetical Sensor Probability Matrix

Actual Target Type	Sensor Indications		
	Friend (F_i)	Hostile (H_i)	Unknown (U_i)
true Friend	A	B	C
true Hostile	D	E	F

Because the sensor can only provide one of three reports, the probability terms in a row must add to one (i.e., A + B + C = 1). To illustrate the process presented in Ralston's paper, consider two ID sensors as shown in Table 2.

Table 2. Example Output Probabilities and States for Two Sensors, A and B

Sensor A

Assumed Target Types	Sensor Output Probabilities		
	Fi-A	Hi-A	U-A
true Friend (tF)	0.7	0.2	0.1
true Hostile (tH)	0.1	0.6	0.3

Sensor B

Assumed Target Types	Sensor Output Probabilities		
	Fi-B	Hi-B	U-B
true Friend (tF)	0.7	0.1	0.2
true Hostile (tH)	0.1	0.6	0.3

	output state	P(state|tF)	P(state|tH)
a	Fi-A and Fi-B	0.49	0.01
b	Fi-A and Hi-B	0.07	0.06
c	Fi-A and U-B	0.14	0.03
d	Hi-A and Fi-B	0.14	0.06
e	Hi-A and Hi-B	0.02	0.36
f	Hi-A and U-B	0.04	0.18
g	U-A and Fi-B	0.07	0.03
h	U-A and Hi-B	0.01	0.18
i	U-A and U-B	0.02	0.09
	sum of above	1.00	1.00

P(state|tF) = probability output state occurs if the target is a true Friend
P(state|tH) = probability output state occurs if the target is a true Hostile

Sensors A and B are both assumed to provide three different types of target indications.[3] Therefore, a total of nine possible sensor-output combinations can be obtained (i.e., 3 × 3 = 9). If we assume that the two sensors are statistically independent (i.e., if we know the output of one of the sensors, it does not change the probability of an output from the other sensor), the probabilities associated with the possible output combinations are the products of the individual sensor probabilities.[4] Table 2 shows the probability products for the nine possible combinations, which are labeled "a" through "i." Because there are exactly nine possible output combinations, and the combinations are statistically independent, the two columns must sum to one, as shown. Table 2 shows that if the target is truly a friend, then sensor output combination "Fi-A and Fi-B" will occur 49 percent of the time. If the target were truly a hostile, then this sensor output combination would only occur 1 percent of the time. For sensibly designed ID sensors, one would expect that friend indications by two sensors would serve as a strong indicator for friendly targets.

Figure 1 shows a plot of the nine sensor output combinations. Combinations "a" and "e" are noted on the figure. Ideally, we would like to always identify a hostile as a hostile and never identify a friend as a hostile. Unfortunately, none of the nine sensor output-state combinations allows us to achieve this condition. In fact, state combination "e" gets closest to the upper left corner of the plot, but it still fails to declare 64 percent of the hostile targets as hostile. If used in an engagement rule, it also would result in declaring 2 percent of friendly targets as hostile. We are not, however, restricted to looking at single output states. For instance, we could group output-state combinations "e" and "h" to achieve hostile declarations for truly hostile targets 54 percent of the time, but with hostile declarations for friendly targets occurring 3 percent of the time.

To consider the problem in its most general terms, the nine sensor output-state combinations (i.e., "a" through "i") can be grouped in 512 different ways. This number arises because each sensor output-state combination could either be included, or not

[3] Ralston's method can consider any number of sensor outputs. Here, three are used to simplify the discussion. Also, the sensor output names (Fi-A, etc.) are arbitrary and are not used when making ID declarations (as you will see, only the associated probabilities are used in the decision process).

[4] If there is partial statistical dependence between the sensors, then the probability terms for one sensor could conceivably be redefined to represent only its ability to provide statistically independent target information; however, the details of this redefinition process could be complicated.

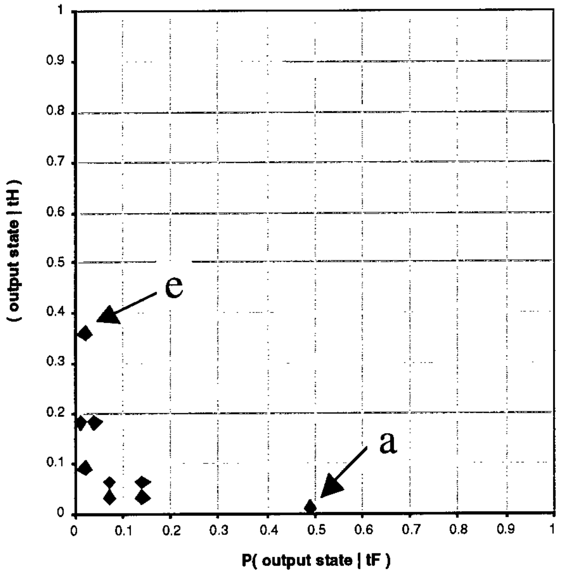

Figure 1. Sensor State Probabilities

included, in any grouping. These two possibilities (either present or not present) for each of the nine sensor output-state combinations create the total of $2^9 = 512$ possible output-state groups. Figure 2 shows the 512 possible output-state groups for the assumed two sensors plotted on a probability-space diagram. Nine of the points shown in Figure 2 are identical to those shown in Figure 1 because some of the groups would include only a single-state combination. The figure also shows an upper and a lower boundary. The two sensors assumed above cannot offer any options between the upper boundary and the desired operating point in the upper left corner of the figure. If additional sensors are added in an attempt to draw the upper boundary nearer to the desired operating point, then the analysis approach presented so far becomes impractical. For example, if a third sensor that also provides friend, hostile, and unknown target indications is added, there are now $3 \times 3 \times 3 = 27$ sensor output states to consider. These 27 sensor output states, however, can be combined in $2^{27} = 134{,}217{,}728$ different ways. Hence, a simpler method of developing the operating curve is needed—that provided in Ralston's paper.

The upper boundary can be developed efficiently by noticing that the boundary starts with a steep slope that gradually decreases. If we were to choose only one of the nine possible outputs, then the first one to choose would be the one with the steepest slope. This slope is actually the ratio of the probability terms. If the nine output states are sorted according to this *likelihood* ratio of probability terms, then the groups of output-state combinations taken in that order define the upper boundary. Table 3 shows the

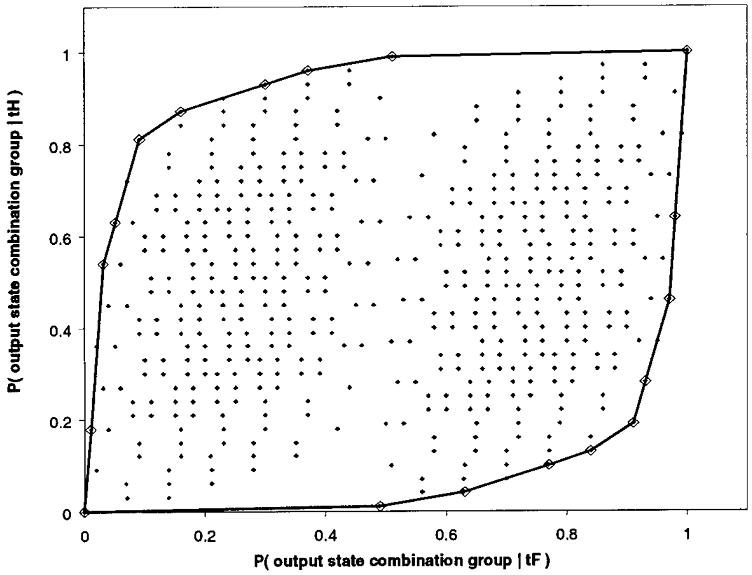

Figure 2. Possible Output-State-Combination Groups

Table 3. Probability Terms Sorted in Descending Order of Their Probability Ratios

	output state	P(state|tF)	P(state|tH)	Ratio	output state	Σ P(state|tF)	Σ P(state|tH)
e	Hi-A and Hi-B	0.02	0.36	18.0	e	0.02	0.36
h	U-A and Hi-B	0.01	0.18	18.0	e + h	0.03	0.54
f	Hi-A and U-B	0.04	0.18	4.5	e + h + f	0.07	0.72
i	U-A and U-B	0.02	0.09	4.5	e + h + f + i	0.09	0.81
b	Fi-A and Hi-B	0.07	0.06	0.8571	e + h + f + i + b	0.16	0.87
d	Hi-A and Fi-B	0.14	0.06	0.4286	e + h + f + i + b + d	0.30	0.93
g	U-A and Fi-B	0.07	0.03	0.4286	e + h + f + i + b + d + g	0.37	0.96
c	Fi-A and U-B	0.14	0.03	0.2143	e + h + f + i + b + d + g + c	0.51	0.99
a	Fi-A and Fi-B	0.49	0.01	0.0204	all nine states	1.00	1.00

probability terms sorted in descending order of their probability ratios. If nine output-state groups are formed by successively adding the next state, then the cumulative sums of the probability output-state combinations shown in the right two columns are obtained. These nine output-state groups, taken in this order, give the points needed to plot the upper boundary[5] in Figure 2. To appreciate the power of this technique, consider the result if a third ID sensor is used. Instead of having to calculate and plot 134,217,728 points, we could form the upper boundary by merely evaluating and ordering 27 probability ratios. Clearly, the upper boundary is of most interest because points (i.e., possible output-state groups) along this boundary are always better than points below and

[5] The lower boundary is formed if the probability ratios are sorted in ascending order. Also, for the specific example given, there are three pairs of output state combinations that have the same probability ratio. Hence, we could optionally first use output state "h" and then add state "e" to obtain the same upper boundary.

to the right of the boundary: we want to operate as close to the upper left corner of the probability space as possible. However, the question of how to select the single best point on this boundary line now arises.

There are at least two strategies for selecting a preferred point on the boundary line. One method applies a predetermined threshold. Another method performs a trade-off between the relative undesirability of allowing enemy leakers versus incurring fratricide of friendly platforms. The threshold option involves setting a criterion such as, "I want to always identify 80 percent of the hostile targets as hostile," and then examining where this falls on the boundary line. In the numerical example presented (see Table 3), this would mean selecting the output-state group "e + h + f + i," which results in 81 percent of the hostile targets being identified as hostile. Thus, if a commander wants to achieve correct identification of at least 80 percent of the hostile targets, and these two ID sensors are the only ones available, then when any of these four output-state combinations occur, the target should be declared hostile. Application of this rule, however, would also result in friends being declared hostile 9 percent of the time.

Now consider the alternative of selecting an operating point by performing a trade-off between the relative undesirability of allowing enemy leakers versus the undesirability of incurring fratricide. This approach entails defining a "cost" expression[6] that represents the relative undesirability of leakers and fratricide. This technique also factors in the force ratio, which represents the possibility that any unknown targets might truly be a friend or a hostile, and leads to the following equation:

$$C_T = C_{pL} \times P_H \times \text{p(hostile not declared hostile)} + C_F \times P_F \times \text{p(friend declared hostile)} \qquad (1)$$

where

C_T = total "cost"

C_{pL} = relative cost of not declaring a hostile as hostile (e.g., a potential leaker)[7]

P_H = a priori probability a yet-to-be-identified target is hostile[8]

p(hostile not declared hostile) = 1 – p(hostile declared hostile),

[6] This means a "cost" expression in the operations research sense. Although the cost of a friendly platform and the cost of a target that the enemy might destroy could be used to define the expressions, we need not perform such detailed accounting. We could set cost values based on instinctive factors such as the relative impact of leakers and fratricide on the ability to win a war quickly. Such a cost approach was presented in Ralston's paper.

[7] While this is not strictly speaking a potential leaker, the subscript is suggestive of the effect and it provides a compact notation.

[8] Proportional to the relative number of hostile targets, N_H. Hence, $P_H \sim N_H$ and $P_F \sim N_F$.

where we also can write

p(hostile declared hostile) as p(output state | tH)

C_F = relative cost of declaring a friend as hostile (e.g., a fratricide)[9]

P_F = a priori probability a yet-to-be-identified target is friendly[10]

p(friend declared hostile) = p(output state | tF).

The probability factors in the above equation are expressed in terms of the output probabilities for the ID sensors—which are unknowns that need to be determined. The first probability factor, p(hostile not declared hostile), equals one minus the probability that a hostile *is* declared hostile. Evaluating Equation 1 for each of the nine output-state groups defined for the example given previously produces the following table of values (note the use of number of hostile and friend targets, N_H and N_F, instead of P_H and P_F).

Table 4. Costs Associated with Output State Groups

output state group	Σ P(state|tF)	Σ P(state|tH)	C_T
e	0.02	0.36	0.84
e + h	0.03	0.54	0.76
e + h + f	0.07	0.72	0.98
e + h + f + i	0.09	0.81	1.09
e + h + f + i + b	0.16	0.87	1.73
e + h + f + i + b + d	0.30	0.93	3.07
e + h + f + i + b + d + g	0.37	0.96	3.74
e + h + f + i + b + d + g + c	0.51	0.99	5.11
all nine states	1.00	1.00	10

C_T evaluated with C_{pL} = 1, C_F = 5, N_H = 1, and N_F = 2

The relative cost, C_T, is minimum for the output group that involves output-state combinations "e" and "h" (see Table 4). Hence, for the assumed values of C_{pL}, C_F, N_H and N_F, the best (i.e., lowest cost) option would be to declare a target hostile only if either ID sensor output-state combinations "e" or "h" occur. Of course, different values for C_{pL}, C_F, N_H and/or N_F could change the location of the minimum cost value.

As a lead in to the multiple-true-target-type case to be presented shortly, we now look at a geometric interpretation of this evaluation. If *x* and *y* are substituted for the probability terms p(output state | tF) and p(output state | tH) and N_H and N_F replace P_H and P_F, then Equation 1 becomes

[9] Declaring a friend to be hostile does not equate one-for-one with fratricide, but it is suggestive of the effect.

[10] Proportional to the relative number of friendly targets, N_F. Hence, $P_H \sim N_H$.

$$y = \left(\frac{C_F N_F}{C_{pL} N_H}\right) x + \left(\frac{C_{pL} N_H - C_T}{C_{pL} N_H}\right) \qquad (2)$$

This is merely the equation for a straight line with a slope of $C_F N_F / C_{pL} N_H$ and intercept $(C_{pL} N_H - C_T)/(C_{pL} N_H)$. As shown in Figure 3, when the term C_T equals zero, the line passes through the point $x = 0$ and $y = 1$, as should be expected because the ideal operating point corresponds to the upper left corner of Figure 2 where p(state | tF) = 0 and p(state | tH) = 1. Figure 3 also shows how the location of the line would vary as C_T increases (i.e., at less desirable positions in the $x - y$ probability space). If Equation 2 is plotted for the previously assumed values of C_{pL}, C_F, N_H, and N_F, and if the upper boundary line is also plotted, then the situation depicted in Figure 4 is obtained. The plot shows the optimum point on the boundary corresponding to the output group "e" and "h." Note that the optimum point is the point on the boundary curve that is closest to the zero-cost line. The figure also shows the cost line for cost = 0.76 that parallels the zero-cost line and passes through the "e + h" output-state group.

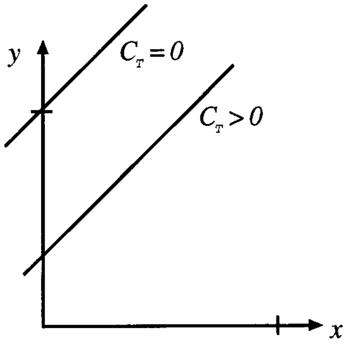

Figure 3. Cost Lines

The geometry in Figure 4 provides a simple basis for locating the minimum-cost point on the boundary line. For the example parameters chosen, the slope of the cost lines equal 10. Therefore, any output-state combination with a probability ratio greater than or equal to 10 should be selected. As shown in Table 3, the ratios p(state | tH)/p(state | tF) for the output combinations "e" and "h" both equal 18—which is greater than the value of 10 derived from the cost equation. All other output-state combinations have values less

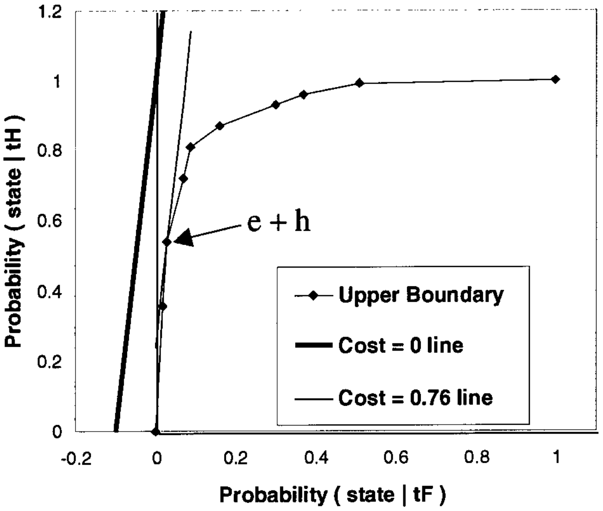

Figure 4. Location of Minimum Cost Point

than 10. Hence, applying the simple ratio test leads to selection of the same preferred operating point that we would obtain either by selecting the minimum cost point along the boundary curve or by finding the point on the boundary that is closest to the zero-cost line.[11]

[11] In the above discussion, the cost method has only selected a single hostile-declaration operating point on the boundary curve. In Ralston's paper, additional operating points were also considered. These additional operating points could be defined by merely introducing alternative values in place of the terms C_{pL} and/or C_F in Equation 2. For example, if the cost of a possible leaker, C_{pL}, were replaced by C_{pH} (the cost of failing to declare a hostile target to be a potential hostile), then a cost line with a different slope would be defined. The point where a line with the slope of this second cost line touches the boundary would define the boundary between making a *potential* hostile declaration and no declaration.

III. ALTERNATE ANALYSIS METHOD

For the probability analysis problems described so far (i.e., picking the best way to combine ID sensor data when only two types of targets are considered), the analysis methods presented above work quite nicely. They break down, however, when a third type of target is introduced into the mix. Therefore, we introduce an alternative approach to solving the two-target-type problem that sets the stage for analyzing problems involving more than two target types.

The first step entails representing each of the points in Figure 1 as a vector in a 2–D probability space. For example, points "a" and "e" in Figure 1 can be expressed as follows:

$$\vec{V_a} = \begin{pmatrix} 0.49 \\ 0.01 \end{pmatrix} \quad \text{and} \quad \vec{V_e} = \begin{pmatrix} 0.02 \\ 0.36 \end{pmatrix} \ ,$$

where

$\vec{V_a}$ = a vector from the origin to the point (0.49,0.01) in probability space, and
$\vec{V_e}$ = a vector from the origin to the point (0.02,0.36) in probability space.

In Figure 5, the vector V_e points towards the cost line, while the vector V_a points away. This suggests an alternative technique for finding which of the individual output-state combinations to include in the preferred operating point—namely,

Include only those output-state combinations that approach, or parallel, the minimum cost line.

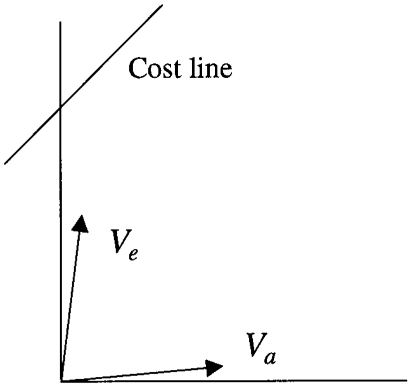

Figure 5. Probability Vectors

For cost lines having a slope of 10, only output-state combinations "e" and "h" point toward the line. Therefore, the group having these two output-state combinations gets closest to the minimum cost line. Linear algebra provides a convenient way to implement this geometric process. A vector perpendicular (i.e., normal) to the cost line for the cost values previously given can be readily computed as follows:

$$\overrightarrow{V_N} = \begin{pmatrix} C_F N_F \\ -C_{pL} N_H \end{pmatrix} = \begin{pmatrix} 10 \\ -1 \end{pmatrix} . \tag{3}$$

This perpendicular, or normal, vector points downward and to the right. From linear algebra, the dot product[12] of the probability vectors with the normal vector will give a negative value if the probability vectors point toward the cost line. The dot product for a probability vector that parallels the cost line will equal zero, because it will be perpendicular to the normal vector. Table 5 shows the dot products of each of the nine output-state combinations with the normal vector defined above. As expected, states "e" and "h" have negative[13] dot products, while the other seven have positive values.

Table 5. Dot Products of Output-State Combinations

	output state	P(state|tF)	P(state|tH)	Dot Product
a	Fi-A and Fi-B	0.49	0.01	4.89
b	Fi-A and Hi-B	0.07	0.06	0.64
c	Fi-A and U-B	0.14	0.03	1.37
d	Hi-A and Fi-B	0.14	0.06	1.34
e	Hi-A and Hi-B	0.02	0.36	**–0.16**
f	Hi-A and U-B	0.04	0.18	0.22
g	U-A and Fi-B	0.07	0.03	0.67
h	U-A and Hi-B	0.01	0.18	**–0.08**
i	U-A and U-B	0.02	0.09	0.11

Hence, this dot product methodology produces the same result as the one previously obtained.

[12] Consider two vectors defined as follows: $\overrightarrow{V_1} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ and $\overrightarrow{V_2} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$. The vector dot product of the vectors is a scalar (i.e., a number) given by $V_1 \bullet V_2 = x_1 \cdot x_2 + y_1 \cdot y_2$. If the vectors involved are unit vectors (i.e., they have a length = 1), then the dot product equals the cosine of the angle between the vectors. For parallel unit vectors, the dot product equals 1 if they point in the same direction and –1 if they point in opposite directions, because cosine(0°) = 1 and cosine(180°) = –1. The dot product of perpendicular vectors equals 0, because cosine(90°) = 0.

[13] If unit vectors V_{Nu} and V_{Pu} are used instead of V_N and V_P (where V_P is one of the probability vectors), then $V_{Nu} \bullet V_{Pu} = \sin(\Theta - \phi)$ when Θ = slope of cost line and ϕ = slope of the probability vector. (Remember, V_N is perpendicular to the cost line.)

IV. EXTENSION TO MULTIPLE-TARGET-TYPE ANALYSIS

This discussion first covers the case of three target types. It then describes how to analyze problems with any arbitrary number of target types.

As before, we begin with a representation of the sensor probability matrices. To keep the example relatively simple, consider the case of only two sensors, A′ and B′, with assumed probability values given in Table 6. Note that these probability matrices contain three rows of numbers corresponding to the three possible true target types. Also, sensor A′ can only provide friend, hostile, or unknown target indications, but sensor B′ can provide four types of target indications (it adds neutral). Again, the sensors could provide any number of different target indications, but the limited number of indications assumed here will help simplify the numerical examples—there will only be 3 × 4 = 12 output-state combinations to consider.

Table 6. Output Probabilities and States for Two Sensors, A′ and B′

Sensor A′

Assumed Target Types	Sensor Output Probabilities		
	Fi-A′	**Hi-A′**	**U-A′**
true Friend (tF)	**0.7**	**0.2**	**0.1**
true Hostile (tH)	**0.1**	**0.8**	**0.1**
true Neutral (tN)	**0.4**	**0.1**	**0.5**

Sensor B′

Assumed Target Types	Sensor Output Probabilities			
	Fi-B′	**Hi-B′**	**Ni-B′**	**U-B′**
true Friend (tF)	**0.7**	**0.1**	**0.1**	**0.1**
true Hostile (tH)	**0.1**	**0.6**	**0.1**	**0.2**
true Neutral (tN)	**0.1**	**0.1**	**0.6**	**0.2**

	output state	P(state|tF)	P(state|tH)	P(state|tN)
a	Fi-A′ and Fi-B′	0.49	0.01	0.04
b	Fi-A′ and Hi-B′	0.07	0.06	0.04
c	Fi-A′ and Ni-B′	0.07	0.01	0.24
d	Fi-A′ and U-B′	0.07	0.02	0.08
e	Hi-A′ and Fi-B′	0.14	0.08	0.01
f	Hi-A′ and Hi-B′	0.02	0.48	0.01
g	Hi-A′ and Ni-B′	0.02	0.08	0.06
h	Hi-A′ and U-B′	0.02	0.16	0.02
i	U-A′ and Fi-B′	0.07	0.01	0.05
j	U-A′ and Hi-B′	0.01	0.06	0.05
k	U-A′and Ni-B′	0.01	0.01	0.3
l	U-A′ and U-B′	0.01	0.02	0.1
	sum of above	1.00	1.00	1.00

These 12 output-state combinations produce 2^{12} = 4,096 output-state-combination groups. Figure 6 shows the 4,096 possible output-state-combination groups plotted in a three-dimensional (3–D) scatter plot and three separate 2–D scatter plots. The main point of Figure 6 is to illustrate that the problem quickly gets very complex, and creation of a

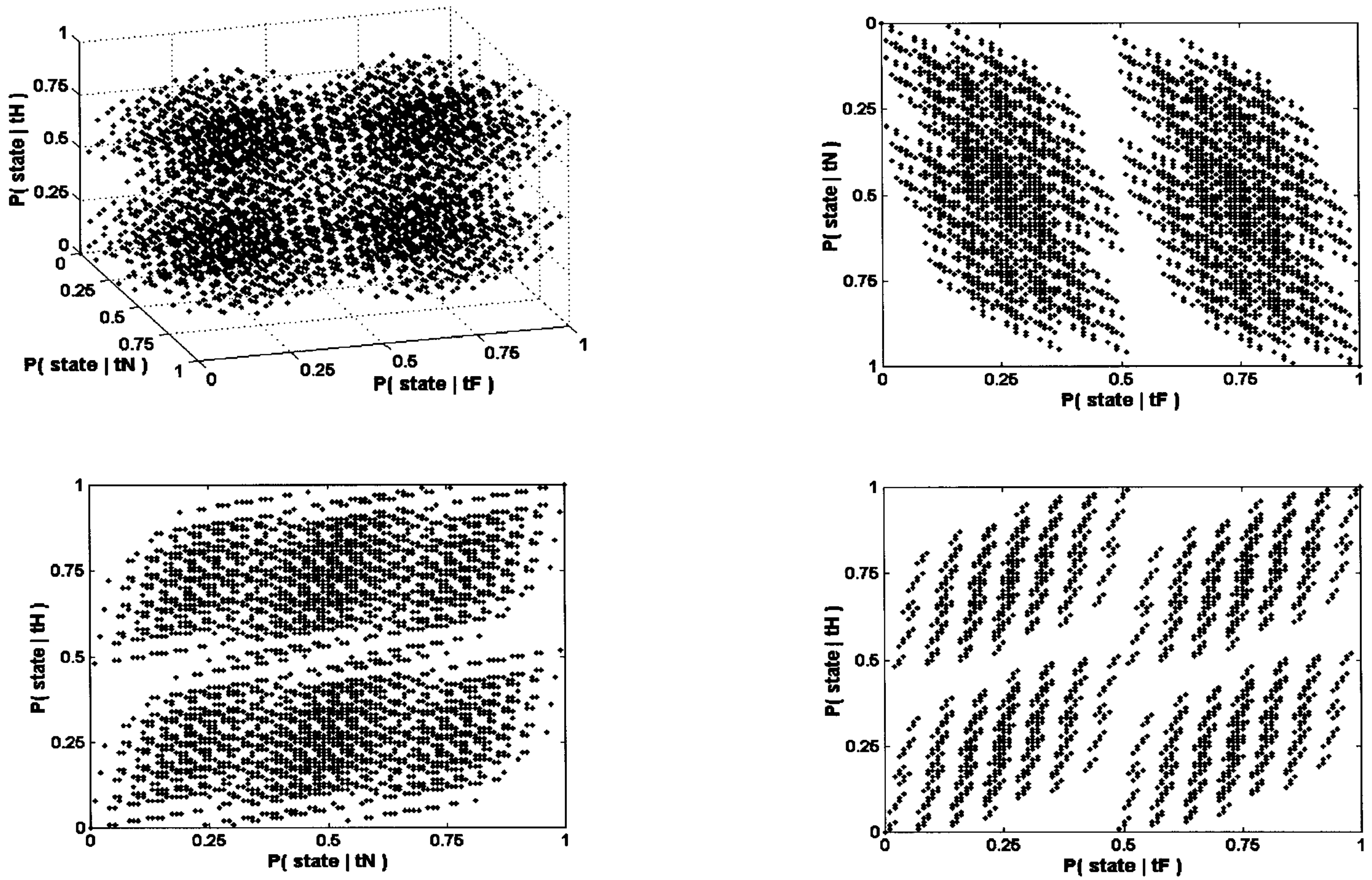

Figure 6. Scatter Plots for 3-D Example

3–D "football" that encloses all the points appears difficult.[14] Just as in the 2–D case, however, we can avoid having to develop the scatter plot and calculate the cost value for each of the points in the scatter plot. We begin by defining a 3–D cost function:

$C_T = C_{pL} \times N_H \times$ p(hostile not declared hostile) + $C_F \times N_F$
$\times$ p(friend declared hostile) + $C_N \times N_N \times$ p(neutral declared hostile) (4)

where

C_T = total "cost"

C_{pL} = relative cost of not declaring a hostile as hostile (e.g., a potential leaker)

N_H = number of hostile targets

p(hostile not declared hostile) = 1 – P(output state | tH)

C_F = relative cost of declaring a friend as hostile (e.g., a fratricide)

N_F = number of friendly targets

p(friend declared hostile) = P(output state | tF)

C_N = relative cost of declaring a neutral as hostile

N_N = number of neutral targets

p(neutral declared hostile) = P(output state | tN)

Substituting (1–*y*), *x*, and *z* for the probability terms in Equation 4 and then reordering the terms yields:

$$C_T = (C_F N_F)x + (C_{pL} N_H) \cdot (1-y) + (C_N N_N)z \quad . \tag{5}$$

Equation 5 defines a cost plane in the 3–D probability space. Ideally, we would like to always declare a hostile to be hostile and never declare a friend or neutral to be hostile. This corresponds to the values $y = 1$ and $x = z = 0$ in the 3–D space. Equation 5 gives the expected result that $C_T = 0$ for these values of *x*, *y*, and *z*. Using reasoning similar to the 2–D case, we want to find the group of output-state combinations that gets closest to this zero-cost plane. To accomplish this, we need only do the following:

- Define a vector that is normal to the cost plane
- Represent each of the output-state combinations as a vector in 3–D probability space (for the example presented, this means defining 12 vectors)

[14] However, one could form a convex hull consisting of only that subset of the 4,096 points that lie on the surface of the 3–D "football" slope. Convex hulls are part of an advanced branch of mathematics called topology, but these details will not be addressed herein.

- Compute the dot product of the vector normal cost surface with the vectors for the output-state combinations
- Identify those output-state combinations that produce a dot product less than, or equal to, 0.

To implement the first of the four steps, a vector normal[15] to the cost plane is calculated as follows:

$$\overrightarrow{V_N} = \begin{pmatrix} C_F N_F \\ -C_{pL} N_H \\ C_N N_N \end{pmatrix} . \tag{6}$$

Assuming $N_F = N_H = N_N = 1$ and $C_F = C_{pL} = C_N = 1$, the corresponding dot products are derived. In this case, four of the output-state combinations pass the dot-product test. The resulting preferred output-state-combination group consists of cases "f," "g," "h," and "j" (see Table 7). This group has a combined probability that would result in declaring 7 percent of friends (i.e., 0.02 + 0.02 + 0.02 + 0.01 = 0.07), 78 percent of hostiles (i.e., 0.48 + 0.08 + 0.16 + 0.06 = 0.78), and 14 percent of neutrals to be hostile.[16] Referring to Figure 6, this point does not appear to lie on any of the boundary lines for the three 2–D views.

[15] Using linear algebra, the normal to the plane can be derived by taking the vector cross product to two nonparallel vectors lying in the cost plane. The length of the normal vector does not matter, so the values shown are given for convenience. The direction of the normal vector *does* matter, so we need to check that the normal vector has a positive dot product with a vector going from the most preferred to the least preferred corners of the probability cube. Alternatively, we could also find the normal vector by rewriting Equation 5 as $C_T = (C_F N_F)\, x + (C_{pL} N_H)\,(1 - y) + (C_N N_N)\, z$ and taking the gradient of C_T. This yields

$$\overrightarrow{V_N} = \nabla C_T = \begin{pmatrix} \partial C_T / \partial x \\ \partial C_T / \partial y \\ \partial C_T / \partial z \end{pmatrix} = \begin{pmatrix} C_F N_F \\ -C_{pL} N_H \\ C_N N_N \end{pmatrix} .$$

[16] Because only 4,096 possible output-state-combinations groups exist for this example, an exhaustive search that calculated the cost function for all 4,096 points was used to confirm this result. In general, however, such an exhaustive search would be extremely impractical.

Table 7. Dot Products of Output-State Combinations for the Three-Target-Type Example

	output state	P(state|tF)	P(state|tH)	P(state|tN)	Dot Products
a	Fi-A′ and Fi-B′	0.49	0.01	0.04	0.52
b	Fi-A′ and Hi-B′	0.07	0.06	0.04	0.05
c	Fi-A′ and Ni-B′	0.07	0.01	0.24	0.30
d	Fi-A′ and U-B′	0.07	0.02	0.08	0.13
e	Hi-A′ and Fi-B′	0.14	0.08	0.01	0.07
f	Hi-A′ and Hi-B′	0.02	0.48	0.01	**−0.45**
g	Hi-A′ and Ni-B′	0.02	0.08	0.06	**0.00**
h	Hi-A′ and U-B′	0.02	0.16	0.02	**−0.12**
i	U-A′ and Fi-B′	0.07	0.01	0.05	0.11
j	U-A′ and Hi-B′	0.01	0.06	0.05	**0.00**
k	U-A′ and Ni-B′	0.01	0.01	0.3	0.30
l	U-A′ and U-B′	0.01	0.02	0.1	0.09
	sum of above	1.00	1.00	1.00	

Extension of this technique to more than three true target types is now straightforward. We need only define a cost function for the various target types, compute the normal vector to this multidimensional (M–D) surface, calculate dot products of the normal vector with vector representations of the possible output-state combinations, and determine which dot products result in a value less than or equal to zero. Considering the four-dimensional (4–D) case with a cost function given by

$$C_T = \alpha w + \beta x + \gamma(1-y) + \delta z \quad . \tag{7}$$

where

w = probability a coalition force participant is declared hostile

x = probability a friend is declared hostile

y = probability a hostile is declared hostile

z = probability a neutral is declared hostile.

The resulting normal vector to this 4–D cost surface becomes[17]

$$\overrightarrow{V_N} = \begin{pmatrix} \alpha \\ \beta \\ -\gamma \\ \delta \end{pmatrix} \quad . \tag{8}$$

[17] To calculate this normal vector, use the gradient method or define three nonparallel vectors that lie in the cost plane and compute their cross product. Ensure that the dot product of the resulting normal vector has a positive dot product with a vector pointing from the point (0, 0, 1, 0) to the point (1, 1, 0, 1).

V. REFINEMENT TO THE COST METHOD

The 2–D, 3–D, and M–D cost methods described above have focused only on the problem of making hostile declarations. A similar technique (with one refinement to be described later) could be applied to make friend, neutral, etc., target declarations. For ease of discussion, refer to the 2–D case presented in Figure 2. We want to declare a friend as a friend and never declare a hostile as a friend. This statement corresponds to the lower right corner of Figure 2. As before, a "cost" expression could be used to pick the preferred point off the *lower* boundary that is "closest" to the desired operating point (i.e., the lower right corner). The cost expression could be formed similar to Equation 1:

$$C_{Tf} = C_{pF} \times N_F \times \text{p(friend not declared friend)} + C_{Lf} \times N_H \times \text{p(hostile declared friend)} \qquad (9)$$

where

C_{Tf} = total "cost" when making friend ID declarations

C_{pF} = relative cost of not declaring a friend as friend (e.g., a potential fratricide)

N_F = number of friendly targets

p(friend not declared friend) = 1 – P(output state | tF)

C_{Lf} = relative cost of declaring a hostile a friend (e.g., an incorrect friend, or probable leaker)

N_H = number of hostile targets

p(hostile declared friend) = P(output state | tH).

As before, let p(output state | tF) = $x'$and p(output state | tH) = y', and Equation 9 becomes[18]

$$y' = \left(\frac{C_{pF} N_F}{C_{Lf} N_H} \right) x' + \left(\frac{C_{Tf} - C_{pF} N_F}{C_{Lf} N_H} \right) \quad . \qquad (10)$$

As expected, when $C_{Tf} = 0$, the cost line passes through $x' = 1$ and $y' = 0$, which is the lower right corner of Figure 2. As C_{Tf}increases, the cost line moves upward and to the left. To use the vector dot-product method described previously, we would need to define

[18] The x', y'notation is used to indicate that these probability values are now associated with making a friend ID declaration (as opposed to the x, y notation associated with hostile ID declarations).

a normal to this new cost line, which points generally upward and to the left. This vector would be

$$\overrightarrow{V_{N_F}} = \begin{pmatrix} -C_{pF}N_F \\ C_{Lf}N_H \end{pmatrix} \quad . \tag{11}$$

Dot products of the vector representations of the output-state combinations with this normal vector would be negative or zero for those output-state combinations included in the minimum-cost output-state-combination group. Similar formulations will work for the 3– or M–D cases.

As indicated above, a refined analysis is required to obtain the completely general solution to the ID problem. Cost Equations 1 and 2 address the total cost associated with hostile ID declarations, while cost Equations 9 and 10 address the costs associated with friend ID declarations. However, there is actually a cross coupling between the two types of cost equations that has so far been ignored. Rewriting Equation 1 with the substitutions p(output state | tF) = x and p(output state | tH) = y yields

$$C_T = \alpha x + \beta(1 - y) \quad , \tag{12}$$

where

$$\alpha = C_F \times N_F$$
$$\beta = C_{pL} \times N_H \quad .$$

Similarly, Equation 9 can be rewritten as

$$C_{Tf} = \gamma(1 - x') + \delta y' \quad , \tag{13}$$

where

$$\gamma = C_{pF} \times N_F$$
$$\delta = C_{Lf} \times N_H \quad .$$

To understand the cross coupling, consider the term (1 – y) in Equation 12. This term, intended to represent the fraction of hostile targets that are potential leakers, was formed by subtracting the probability (or fraction) of hostile targets identified as hostile from one. This would be correct if a hostile target is either identified as hostile or not identified; however, hostile targets could also be misidentified as friends (probable leakers). Therefore, the probability of potential leakers should actually be $(1 - y - y')$. Similarly, the $(1 - x')$ term in Equation 13 should become $(1 - x' - x)$. With these adjustments to the terms representing potential leakers and potential fratricide, an overall cost expression, C_O, becomes

$$C_O = K_T + K_{Tf} = \alpha x + \beta(1 - y - y') + \gamma(1 - x' - x) + \delta y' \ , \qquad (14)$$

which can be rewritten as

$$C_O = K_T + K_{Tf} = (\alpha - \gamma) x + \beta(1 - y) + \gamma(1 - x') + (\delta - \beta) y' \ , \qquad (15)$$

where

$K_T = (\alpha - \gamma) x + \beta(1 - y)$, the modified version of equation 12, and

$K_{Tf} = \gamma(1 - x') + (\delta - \beta) y'$, the modified version of equation 13.

This analysis leads to the realization that the cross coupling just described can be addressed by merely modifying the cost equation coefficients and proceeding as before to determine hostile and friend declarations.[19] Hence, to consider the cross-coupling effect for the two-target type problem, the normal vector for hostile target declarations (as given previously by Equation 3) should now become

$$\overrightarrow{V_{N_H}} = \begin{pmatrix} C_F N_F - C_{pF} N_F \\ -C_{pL} N_H \end{pmatrix} \ . \qquad (16)$$

Also, the modified normal vector for evaluating friend declarations should become

$$\overrightarrow{V_{N_F}} = \begin{pmatrix} -C_{pF} N_F \\ C_{Lf} N_H - C_{pL} N_H \end{pmatrix} \ . \qquad (17)$$

Similar adjustments to the normal vector components would apply to the three- and M-target-type problem.

[19] We could also choose to deal with Equation 14 directly and treat the two-target-type problem with two types of target declaration decisions as a 4–D problem involving the four variables—x, x', y, and y'. The normal to Equation 14 would be found using the gradient; however, the probability space vectors would have to be redefined to have four components formed by taking all possible pairings of (x, y) with (x', y'). Although possible, this seems more complicated than the approach given here.

VI. IMPLEMENTATION CONSIDERATIONS

The technique described above can be easily implemented so long as ressonable estimates of the ID sensor probability matrices are available. However, it also requires a commander to establish the relative "cost" of a fratricide, leaker, misidentified neutrals, etc. The commander could allow these cost parameters to vary throughout the theater (e.g., leakers near troop concentrations might be less tolerable than leakers in remote areas). The commander might also want to vary the cost parameters as a function of the progress of the conflict. For example, before hostilities, the cost of a fratricide or a misidentified neutral could be high, but after initiation of hostilities the relative cost of a leaker could increase.

Once the cost function is defined, the normal to that cost surface can be computed using Equations 3, 6, 8, 16, 17 or similar higher order formulations, depending upon the nature of the ID problem to be analyzed. This is a one-time calculation for any given set of cost function coefficients. The normal vector need only be re-computed when and where the cost parameter(s) change. When a set of ID sensors make an observation, each sensor will need to provide a probability estimate associated with its observation. Although ID sensor probability estimates are not typically reported today, such reports should be relatively easy to implement. The probability estimates (i.e., the probability this observation would occur if the target is actually a friend, hostile, etc.) can contain typical default values, or they could vary according to the sensor environment[20] (e.g., the sensor's signal-to-noise ratio, the total number of targets in the region). With similar information from all sensors, a probability vector for the combined set of ID observations can be computed. The dot product of this probability vector with the normal vector will determine if this set of observations should result in a hostile declaration. Somewhat surprisingly, this approach avoids the necessity of ever having to compute the optimum

[20] This is analogous to a radar providing both an estimated target position *and* a covariance matrix.

output-sensor-combination group.[21] Rather, as new ID data arrives, or as the probability estimates for ID reports change, a hostile target declaration decision can be made on the fly by merely recomputing the dot product using the newly available probability vector.

The approach described above has focused on the determination of hostile target declarations, but if we formed a cost function associated with friend, neutral, etc., declarations, a similar procedure could be employed. Clearly, the technique described herein is needed if we wish to use Bayesian methods to make three or more types of target declarations. On the other hand, there may be instances where there are three or more types of targets, but the need is only to declare targets as either engageable or not. In this case, it can be argued that the methods presented herein are unnecessary because only two types of target declarations are being made. The appendix provides a mathematical demonstration that the multiple-target-type problem can be dealt with as a two-target-type problem (by combining all the engageable target types into one category and all non-engageable target types into a second category) under *only one* special case. The special case requires (1) that the cost weightings for *all* the engageable targets be the same and (2) that the cost weightings for *all* the non-engageable targets be the same. If this is not true, then doing an analysis of a multiple-target-type problem as a two-target-type problem will lead to different target declarations.

[21] To determine whether an ID sensor output combination passes the "cost" function test, it is not necessary to rank order the possible output sensor combinations to develop, for example, the upper boundary shown in Figure 2. This curve could be developed, however, as part of the decision process associated with initially setting the cost function coefficients and understanding the overall percentage of hostile, friendly, etc., targets that would be engaged if a proposed cost rule were adopted. Also, if a threshold rule (e.g., ensure engagement of 85 percent of all hostile targets) were used instead of a cost rule, then the boundary curve would have to be generated. In this latter case, the output-state-combination group that best meets the threshold would have to be established before a conflict; during the conflict, the ID process would have to determine whether a specific output-state combination was a member of the pre-established output-state-combination group. Trying to make a threshold rule adapt to variations in ID sensor performance as the conflict progresses would likely prove unwieldy.

VII. CONCLUSIONS

This paper provides a mathematical formulation for determining the optimum combination of ID sensor output states when three or more types of true targets must be considered. Although the mathematical reasoning may seem nontrivial, the resulting technique involves mathematically trivial calculations to determine if a set of ID sensor reports should result in a hostile declaration. The methodology requires a paradigm shift from current fixed-ID rules of engagement. Commanders would have to express the relative importance of fratricide, leakers, etc., in terms of a cost function. Also, either default or adaptive ID sensor probability values would have to be provided as a part of ID sensor reports. These modifications appear worth serious consideration, given that we typically fight in complex theaters that involve many types of targets, and there is always a premium on making the best possible ID decisions.

APPENDIX

APPENDIX

The following demonstrates that the multiple-target-type problem can be analyzed as a two-target-type problem in the declaration process, without loss of accuracy, only when a very special situation occurs. Consider a four-target type problem, and assume that target types 1 and 2 should both be engaged (e.g., bombers, cruise missiles), while neither target type 3 nor 4 should be engaged.

n_1, n_2, n_3, and n_4 are the numbers for the four target types

p_1, p_2, p_3, and p_4 are the elements of the resulting probability vector for some combination of reports by the various ID sensors

$-\alpha$, $-\beta$, γ, and δ are the relative cost coefficients when analyzing the problem as a 4-target-type problem (the negative signs indicate that one wants to declare target types 1 and 2)

$-a$ and b are the relative cost coefficients when analyzing the problem as a 2-target-type problem

When evaluating the probability vector using the 4-target-type problem approach, the resulting dot product becomes

$$\mathrm{DP}_4 = -\alpha\, n_1\, p_1 - \beta\, n_2\, p_2 + \gamma n_3\, p_3 + \delta\, n_4\, p_4 \quad .$$

To reduce the 4-target-type problem to a 2-target-type problem, assume that the two probabilities for the engageable targets (and the two probabilities for the non-engageable targets) can be combined as follows:

$$p_{\text{engageable}} = \frac{n_1 p_1 + n_2 p_2}{n_1 + n_2} \qquad\qquad p_{\text{non-engageable}} = \frac{n_3 p_3 + n_4 p_4}{n_3 + n_4} \quad .$$

The resulting dot product for the 2-target-type problem approach becomes

$$\mathrm{DP}_2 = -a\ (n_1 + n_2)\ p_{\text{engageable}} + b\ (n_3 + n_4)\ p_{\text{non-engageable}} \quad \text{, or}$$

$$\mathrm{DP}_2 = -a\ n_1\, p_1 - a\ n_2\, p_2 + b\ n_3\, p_3 + b\ n_4\, p_4 \quad .$$

To have $\mathrm{DP}_4 \equiv \mathrm{DP}_2$ for all possible values of p requires $\alpha = \beta = a\ and\ \gamma = \delta = b$. Therefore, unless the cost penalty associated with all engageable targets and the cost penalty for all non-engageable targets are equal, a multiple-target-type problem should not be reduced to a 2-target-type problem.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 2000

3. REPORT TYPE AND DATES COVERED
Final—October 1999–November 1999

4. TITLE AND SUBTITLE
Optimum ID Sensor Fusion for Multiple Target Types

5. FUNDING NUMBERS
DASW01 98 C 0067
CA-2-1617

6. AUTHOR(S)
J. Kent Haspert

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Document D-2451

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
JTAMDO
Crystal Mall 3, Suite 511
Arlington, VA 22202

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 180 words)*

Bayesian techniques have been employed to combine, or fuse, reports from multiple target identification (ID) sensors. Recent work has shown that a "cost" criterion can be used to make optimum target declaration decisions from the Bayesian representations of the ID sensors. However, the "cost" method, as previously formulated, only works when choosing between two types of target declarations (e.g., friend vs. hostile). This paper provides an alternative formulation to the cost methodology previously developed for Bayesian analysis of ID sensor fusion. This alternative formulation is then used to develop a formulation that works for multiple types of targets. Therefore, this alternative formulation will allow analysts and military commanders to include neutral targets in the ID fusion process. Indeed, the technique also enables optimum ID sensor fusion when considering multiple additional types of targets (e.g., coalition force, hostile fighters, hostile bombers, and hostile cruise missiles).

14. SUBJECT TERMS
Bayesian, sensor fusion, optimization, identification, multiple target types

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR